



KEMNAKER



Fundamental JavaScript: Control Flow, Data Structure & Function





KEMNAKER

sanbercode



Outline

- Conditional Javascript
- Looping Javascript
- Array Javascript
- Function Javascript
- Object Javascript





Conditional Javascript





KEMNAKER



Conditional Javascript

Conditional merupakan cara yang digunakan dalam program komputer untuk mengambil keputusan dari kemungkinan benar(true) atau salah(false) terhadap beberapa kondisi.





KEMNAKER



if

Contoh 1 menjalankan kode jika premis bernilai true

```
if ( true ) {  
  console.log("jalankan code")  
}
```

Contoh 2 kode tidak dijalankan jika premis bernilai false

```
if ( false ) {  
  console.log("code tidak dijalankan")  
}
```

Contoh 3 Premis dengan perbandingan suatu nilai

```
var mood = "happy"  
if ( mood == "happy" ) {  
  console.log("hari ini aku bahagia!")  
}
```





KEMNAKER



if else dan if else if

Contoh 1 Branching sederhana

```
var minimarketStatus = "open"
if ( minimarketStatus == "open" ) {
  console.log("saya akan membeli telur dan buah")
} else {
  console.log("minimarketnya tutup")
}
```

Contoh 2 Branching dengan kondisi

```
var minimarketStatus = "close"
var minuteRemainingToOpen = 5
if ( minimarketStatus == "open" ) {
  console.log("saya akan membeli telur dan buah")
} else if ( minuteRemainingToOpen <= 5 ) {
  console.log("minimarket buka sebentar lagi, saya tungguin")
} else {
  console.log("minimarket tutup, saya pulang lagi")
}
```





Nested if

Contoh percabangan bersarang

```
var minimarketStatus = "open"
var telur = "soldout"
var buah = "soldout"
if ( minimarketStatus == "open" ) {
  console.log("saya akan membeli telur dan buah")
  if(telur == "soldout" || buah == "soldout" ) {
    console.log("belanjaan saya tidak lengkap")
  } else if( telur == "soldout") {
    console.log("telur habis")
  } else if ( buah == "soldout" ) {
    console.log("buah habis")
  }
} else {
  console.log("minimarket tutup, saya pulang lagi")
}
```





Switch Case

contoh switch case

```
var buttonPushed = 1;
switch(buttonPushed) {
  case 1: { console.log('matikan TV!'); break; }
  case 2: { console.log('turunkan volume TV!'); break; }
  case 3: { console.log('tingkatkan volume TV!'); break; }
  case 4: { console.log('matikan suara TV!'); break; }
  default: { console.log('Tidak terjadi apa-apa'); }
}
```





KEMNAKER



Ternary Operator

ternary operator merupakan pengondisian yang bisa di buat dengan satu baris

```
// tanpa ternary operator
var age = 20
var bisaVote = ""
if (age > 20){
  bisaVote = "bisa vote"
}else{
  bisaVote = "belum bisa vote"
}
console.log(bisaVote)

// dengan ternary Operator
var age= 20
var bisaVote= age > 20 ? "bisa vote" : "belum bisa vote" // hasilnya belum bisa vote
console.log(bisaVote)
```



KEMNAKER



Ternary Operator

```
// tanpa ternary operator
var angka = 9
var jenisBilangan = ""
if (angka % 2 === 0){
  jenisBilangan = "Bilangan Genap"
}else{
  jenisBilangan = "Bukan Bilangan Genap"
}
console.log(jenisBilangan)

// dengan ternary Operator
var angka = 9
var jenisBilangan = angka % 2 === 0 ? "Bilangan Genap" : "Bukan Bilangan Genap" //hasilnya bukan bilangan genap
console.log(jenisBilangan)
```





Looping Javascript





Looping di Javascript

Loop/Iteration adalah tindakan mengulang / merepetisi sebuah proses, dengan tujuan untuk mendapatkan deret hasil, atau dengan tujuan mendapatkan hasil tertentu dengan repetisi. Setiap proses repetisi ini disebut sebagai Iteration atau Looping.

Untuk melakukan looping/iteration, JavaScript menyediakan beberapa jenis iteration, yaitu:

- **for-loop**
- **while-loop**





For-loop

For-loop adalah bentuk iterasi dengan statement for menjadi kontrol atas loop yang dilakukan.

Berikut adalah cara atau struktur penulisan for-loop:

```
for([Inisialisasi]; [Kondisi]; [Incremental/Decremental]) {  
    [Proses] // Merupakan proses yang akan dijalankan dalam satu iterasi  
}
```





KEMNAKER



Contoh For-loop

contoh 1

```
for(var angka = 1; angka < 10; angka++) {  
  console.log('Iterasi ke-' + angka);  
}
```

contoh 2

```
var jumlah = 0;  
for(var deret = 5; deret > 0; deret--) {  
  jumlah += deret;  
  console.log('Jumlah saat ini: ' + jumlah);  
}  
  
console.log('Jumlah terakhir: ' + jumlah);
```





KEMNAKER



While-loop

While-loop adalah iterasi yang akan mengulang proses dengan kondisi tertentu.

Berikut adalah cara atau struktur penulisan while-loop terdapat dua cara:

```
//cara 1
while([Kondisi]) {
|   [Proses]
}

//cara 2
do {
|   [Proses]
} while([Kondisi])
```





KEMNAKER



Contoh While-loop

contoh 1

```
var flag = 1;
while(flag < 10) {
  console.log('Iterasi ke-' + flag);
  flag++;
}
```

contoh 2

```
var flag = 9;
do {
  console.log('Iterasi ke-' + flag);
  flag--;
} while(flag > 0)
```





KEMNAKER



Waspadai Infinite Loop

```
var flag = 1;  
while(flag < 10) {  
  console.log('Iterasi ke-' + flag);  
}
```

jika kode diatas dijalankan maka akan menghasilkan:

```
Iterasi ke-1  
Iterasi ke-1  
Iterasi ke-1  
Iterasi ke-1  
Iterasi ke-1
```





KEMNAKER



Sesi Tanya Jawab





Array Javascript





Array Javascript

Array adalah kumpulan atau tumpukan berbagai data. Cara menuliskan array yaitu dengan kurung siku ([]) dan elemen-elemen nya dipisah menggunakan tanda koma (,). Setiap elemen dari array memiliki indeks yang dimulai dari 0, 1, 2, dan seterusnya.

berikut ini ilustrasi dari Array:

[0]	[1]	[2]	[3]	[4]
2	5	1	3	4

berdasarkan ilustrasi diatas maka kode array akan seperti dibawah ini:

```
var angka = [2, 5, 1, 3, 4]
console.log(array[0])
console.log(array[2])
```





KEMNAKER



Array Method

Berikut ini adalah beberapa metode atau built-in functions yang dimiliki oleh tipe data array

- **length:** menentukan panjang data dari array
- **push:** menambah 1 nilai ke array ke index paling belakang
- **pop:** menghapus 1 nilai dari array index paling belakang
- **unshift:** menambah 1 nilai ke array index paling depan (index 0)
- **shift:** menghapus 1 nilai dari array index paling depan (index 0)
- **join:** menggabungkan seluruh element array menjadi sebuah string dan mengambil parameter sebagai simbol penyambung antar elemen
- **split:** memecah string dan mengembalikan array sesuai dengan separator / pemisah yang didefinisikan
- **sort:** mengurutkan elemen di dalam array sesuai alphabet
- **slice:** mengambil beberapa lapis data
- **splice:** mengubah nilai array dengan menghapus dan/atau menambah nilai baru ke array





KEMNAKER



Looping Array

Berikut ini contoh looping dengan Array:

```
// contoh 1
var arr = ['Jeruk', 'Mangga', 'Apel'];
for(var i = 0; i < arr.length; i++) {
  console.log(arr[i]);
}

// contoh 2
var names = ["john", "doe", "jack"]
var i = 0
while (i < names.length) {
  console.log(arr[i])
  i++
}
```





Function Javascript

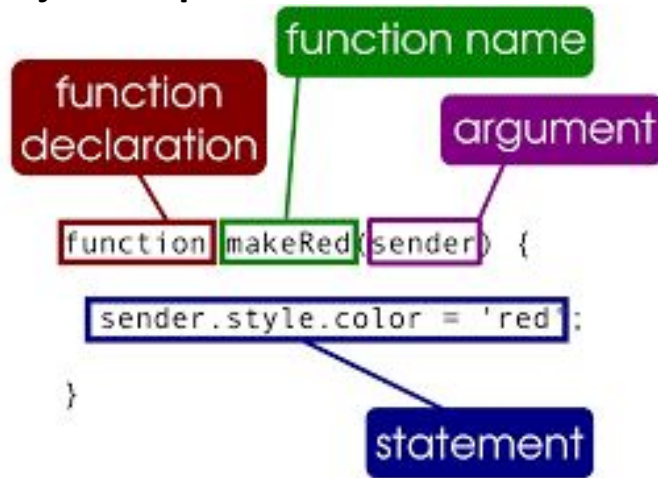




Function Javascript

Function adalah sebuah blok kode yang disusun sedemikian rupa untuk menjalankan sebuah tindakan. Blok kode ini dibuat untuk dapat bisa digunakan kembali.

Perhatikan ilustrasi function javascript dibawah ini :





KEMNAKER



Implementasi Function Javascript

Contoh Function 1: Function sederhana tanpa return

```
function tampilkan() {  
  console.log("halo!");  
}  
  
tampilkan();
```

Contoh Function 2: Function sederhana dengan return

```
function munculkanAngkaDua() {  
  return 2  
}  
  
var tampung = munculkanAngkaDua();  
console.log(tampung)
```





KEMNAKER



Implementasi Function Javascript

Contoh Function 3: Function dengan parameter

```
function kalikanDua(angka) {  
  return angka * 2  
}  
  
var tampung = kalikanDua(2);  
console.log(tampung)
```

Contoh Function 4: Pengiriman parameter lebih dari satu dan nilai default

```
function tampilkanAngka(angkaPertama, angkaKedua=2) {  
  return angkaPertama + angkaKedua  
}  
  
console.log(tampilkanAngka(5, 3))  
console.log(tampilkanAngka(6))
```



KEMNAKER



Anonymous Function

Kita juga dapat menampung function sebagai variable dengan sebuah bentuk function yang dinamakan Anonymous Function.

```
var fungsiPerkalian = function(angkaPertama, angkaKedua) {  
    return angkaPertama * angkaKedua  
}  
  
console.log(fungsiPerkalian(2,5))
```





KEMNAKER



Sesi Tanya Jawab





Object Javascript





Object Javascript

Object adalah kumpulan data tidak berurut yang berisikan pasangan property (key) dan value . Jika kita ingat pada tipe data **Array** yang merupakan kumpulan data yang berurut sesuai indeks, sedangkan **Object** mirip dengan **Array** tapi kini kita memberikan property atau key sendiri.

cara menuliskan **Object** menggunakan kurung kurawal (`{}`) berbeda dengan **array** yang menggunakan kurung siku.





Deklarasi Object

berikut ini contoh deklarasi object dalam Javascript:

```
var car = {  
  brand: "Ferrari",  
  type: "Sports Car",  
  price: 50000000,  
  "horse power": 986  
}
```





KEMNAKER



Mengakses Nilai pada Object

selain Object bisa diakses secara keseluruhan, object pun bisa diakses per-key

```
var myMotorCycle = {  
  brand: "Honda",  
  type: "CBR",  
  "price tag": 30000000  
}  
  
console.log(myMotorCycle.brand) // "Honda"  
console.log(myMotorCycle["price tag"]) // 30000000
```




KEMNAKER



Array of Object

object bisa termasuk dalam tipe data yang berarti dapat dimasukkan ke dalam array, seperti contoh di bawah ini

```
var mobil = [  
  {merk: "BMW", warna: "merah", tipe: "sedan"},  
  {merk: "toyota", warna: "hitam", tipe: "box"},  
  {merk: "audi", warna: "biru", tipe: "sedan"}  
]
```





Array Iteration

array iteration merupakan method dalam array untuk melakukan perulangan data dari array, method array iteration ada banyak tapi untuk basic kita hanya perlu menggunakan 3 method ini yaitu `forEach()`, `map()` dan `filter()`





KEMNAKER



.forEach()

.forEach() method untuk array berfungsi untuk perulangan data dari array, misal kita punya array seperti di bawah ini:

```
var mobil = [  
  {merk: "BMW", warna: "merah", tipe: "sedan"},  
  {merk: "toyota", warna: "hitam", tipe: "box"},  
  {merk: "audi", warna: "biru", tipe: "sedan"}  
]  
  
mobil.forEach(function(item){  
  console.log("warna : " + item.warna)  
})
```





KEMNAKER



.map()

.map() method untuk array berfungsi untuk membuat array baru. misal dengan var mobil diatas kita buat kode seperti di bawah ini:

```
var mobil = [  
  {merk: "BMW", warna: "merah", tipe: "sedan"},  
  {merk: "toyota", warna: "hitam", tipe: "box"},  
  {merk: "audi", warna: "biru", tipe: "sedan"}  
]  
  
var arrayWarna = mobil.map(function(item){  
  return item.warna  
})  
  
console.log(arrayWarna)
```



KEMNAKER



.filter()

.filter() method untuk array berfungsi untuk memnyaring data yang diinginkan. misal dengan var mobil diatas kita buat kode seperti di bawah ini:

```
var mobil = [  
  {merk: "BMW", warna: "merah", tipe: "sedan"},  
  {merk: "toyota", warna: "hitam", tipe: "box"},  
  {merk: "audi", warna: "biru", tipe: "sedan"}  
]  
  
var arrayMobilFilter = mobil.filter(function(item){  
  return item.tipe !== "sedan";  
})  
  
console.log(arrayMobilFilter)
```



KEMNAKER

sanbercode



Kesimpulan

- Conditional Javascript
- Looping Javascript
- Array Javascript
- Function Javascript
- Object Javascript





Terima Kasih

